MULTI CHANNEL APPLICATIONS FOR CONTROL SYSTEM STUDIO (CSS)

K. Shroff#, G. Carcassi, BNL, Upton, Long Island, New York, USA
R. Lange, HZB, Berlin, Germany

Abstract

With the development of the ChannelFinder[1][2] directory service and the PVManager[3] client library, a new set of applications have been developed for Control System Studio(CSS)[4]. These applications have simplified user interaction by requiring the user to provide them with only the criteria of the channels they are interested in, instead of the complete set of channels. They have improved performance with the use of PVManager to manage the collection of control system data.

INTRODUCTION

The High level applications tend to prefer an hierarchical view of the control system name space where they can group channel by location, physical function, etc. These applications also require connections to large set of channel access connections to retrieve the value, alarm state etc.

Using the ChannelFinder we are able to provide High level applications with a hierarchical view of the control system with a simple mechanism to create muti-channel connections using PVManager.

ChannelFinder Service

ChannelFinder tries to overcome the flat name space limitation of the EPICS Channel Access protocol by implementing a generic directory service, which applications can query for a list of channels that match certain conditions, such as physical functionality or location. It also provides mechanisms to create channel name aliases, allowing for different perspectives of the same set of channel names.

ChannelFinder directory server is implemented as a REST style web service, with a relation database (RDB) backend. A client sends an HTTP request to the service to query, create, update and delete directory entries. The data (list of channels and their attributes) is sent/returned in XML (or JSON) notation.

PVManager

PVManager is client library aimed to simplify gathering and composition of control system data. When writing a client application for a control system, common elements always have to be implemented such as rate decoupling, queuing, caching, aggregation, notification on the the correct thread. PVManager allows you to assemble your pipeline starting from commonly used elements , instead of re-implementing every time.

*Work supported by U.S. Department of Energy
#shroffk@bnl.gov

PVManager provides support for composite datatypes like MultiChannelArray, MultiChannelMaps, etc. and provides mechanisms to define expressions to govern the manner in which the control system data is collected and presented to the application.

ChannelFinderClient (CFC)

An eclipse plugin which provides various clients to the various CSS applications to query the channelfinder service. The plugin also provides a set of utility methods to simplify the querying of the service and perform some common operations on the result of these queries.

ARCHITECTURE

The figure 1. shows the architecture of the new CSS applications which use the ChannelFinder service along with PVManager and CFC library plugins. The applications use the CFC plugin, to query the service for a group of channels based on set of criteria which might include positional, physical or operations properties and/or tags associated with the channels, the resulting channels are then used by PVManager to create channel access (CA) connections to retrieve their values.

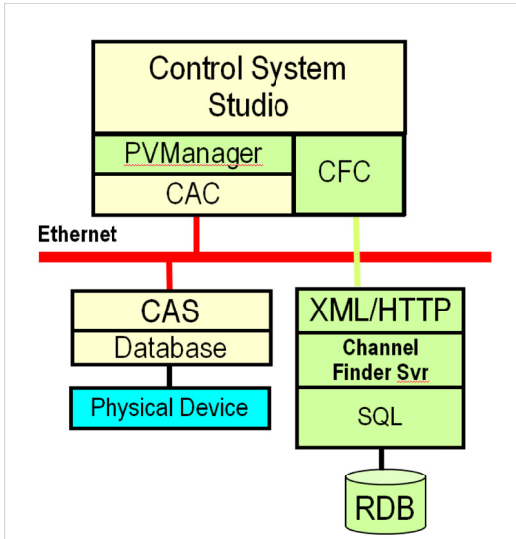
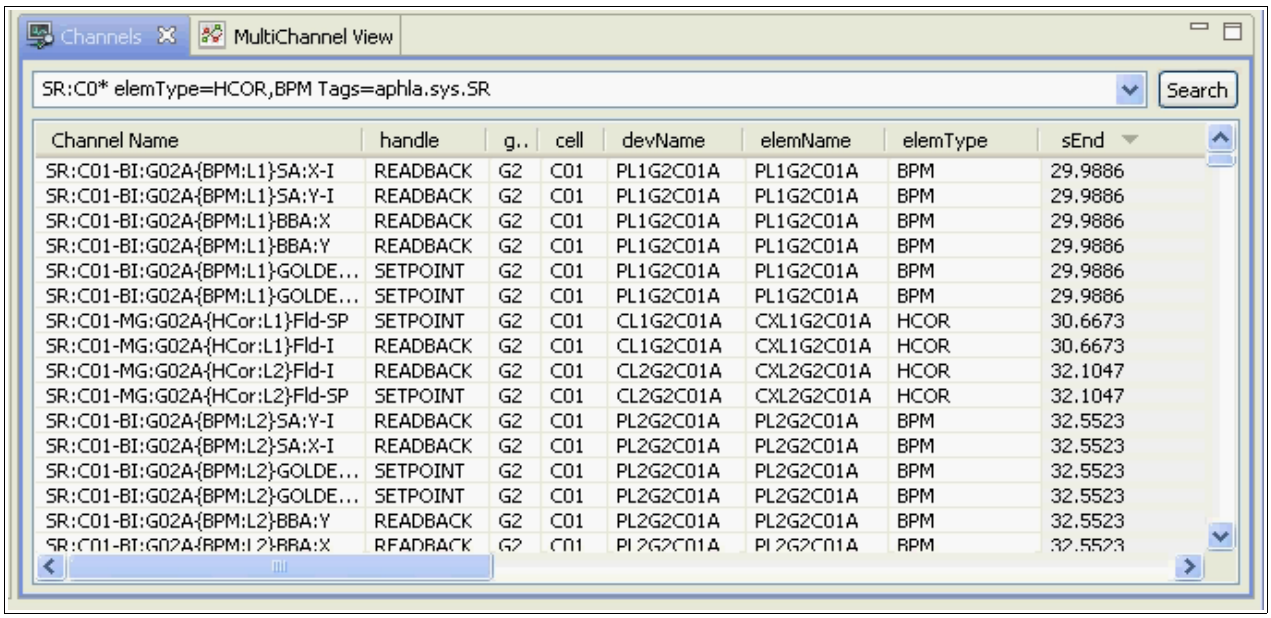


Figure 1: Architecture.

CSS APPLICATIONS

ChannelViewer

The ChannelViewer is a simple graphical application which can be used to query the channel finder. The input



Channel Name	handle	g..	cell	devName	elemName	elemType	sEnd
SR:C01-BI:G02A{BPM:L1}SA:X-I	READBACK	G2	C01	PL1G2C01A	PL1G2C01A	BPM	29.9886
SR:C01-BI:G02A{BPM:L1}SA:Y-I	READBACK	G2	C01	PL1G2C01A	PL1G2C01A	BPM	29.9886
SR:C01-BI:G02A{BPM:L1}BBA:X	READBACK	G2	C01	PL1G2C01A	PL1G2C01A	BPM	29.9886
SR:C01-BI:G02A{BPM:L1}BBA:Y	READBACK	G2	C01	PL1G2C01A	PL1G2C01A	BPM	29.9886
SR:C01-BI:G02A{BPM:L1}GOLDE...	SETPOINT	G2	C01	PL1G2C01A	PL1G2C01A	BPM	29.9886
SR:C01-BI:G02A{BPM:L1}GOLDE...	SETPOINT	G2	C01	PL1G2C01A	PL1G2C01A	BPM	29.9886
SR:C01-MG:G02A{HCor:L1}Fld-SP	SETPOINT	G2	C01	CL1G2C01A	CXL1G2C01A	HCOR	30.6673
SR:C01-MG:G02A{HCor:L1}Fld-I	READBACK	G2	C01	CL1G2C01A	CXL1G2C01A	HCOR	30.6673
SR:C01-MG:G02A{HCor:L2}Fld-I	READBACK	G2	C01	CL2G2C01A	CXL2G2C01A	HCOR	32.1047
SR:C01-MG:G02A{HCor:L2}Fld-SP	SETPOINT	G2	C01	CL2G2C01A	CXL2G2C01A	HCOR	32.1047
SR:C01-BI:G02A{BPM:L2}SA:Y-I	READBACK	G2	C01	PL2G2C01A	PL2G2C01A	BPM	32.5523
SR:C01-BI:G02A{BPM:L2}SA:X-I	READBACK	G2	C01	PL2G2C01A	PL2G2C01A	BPM	32.5523
SR:C01-BI:G02A{BPM:L2}GOLDE...	SETPOINT	G2	C01	PL2G2C01A	PL2G2C01A	BPM	32.5523
SR:C01-BI:G02A{BPM:L2}GOLDE...	SETPOINT	G2	C01	PL2G2C01A	PL2G2C01A	BPM	32.5523
SR:C01-BI:G02A{BPM:L2}BBA:Y	READBACK	G2	C01	PL2G2C01A	PL2G2C01A	BPM	32.5523
SR:C01-BI:G02A{BPM:L2}BBA:X	READBACK	G2	C01	PL2G2C01A	PL2G2C01A	BPM	32.5523

Figure 2: The Channel Viewer can be used to query the ChannelFinder service for a set of channels. The figure shows the result of a query for all the channels in the Storage ring associated with Beam Position Monitor (BPM) or Horizontal Correctors (HCOR) with the tag aphla.sys.SR.

consists of a query can be based one or any combination of the criteria: Channel name, Property values, Tags.

The resulting Channels can be sorted, grouped and tagged. They can be exported to any other CSS application as a set of process variables, thus the end user need not remember details of each process variable but can rather work with logical group determined by the query conditions.

MultiChannel Viewer

The MultiChannel Viewer application shown in Figure 3 is designed to plot the values of a group of channels which are ordered based on a user specified criteria i.e. s-position

The MultiChannel viewer accepts a query (similar to the one used in ChannelViewer) which represents a logical group of channels on the ChannelFinder server.

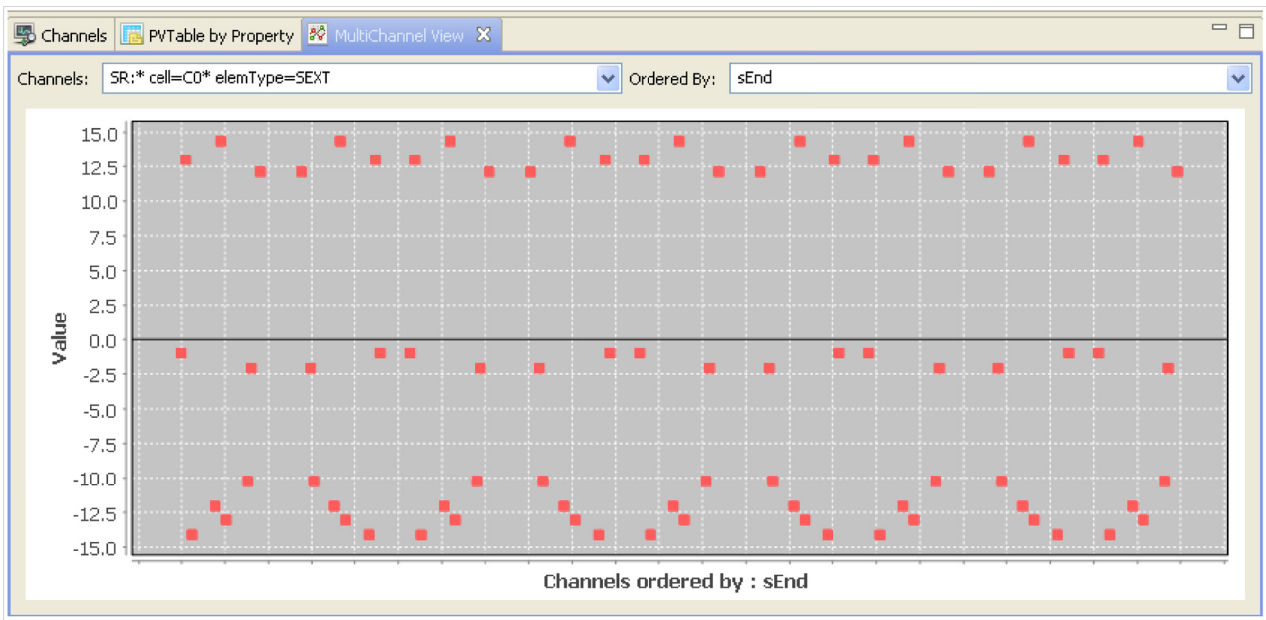
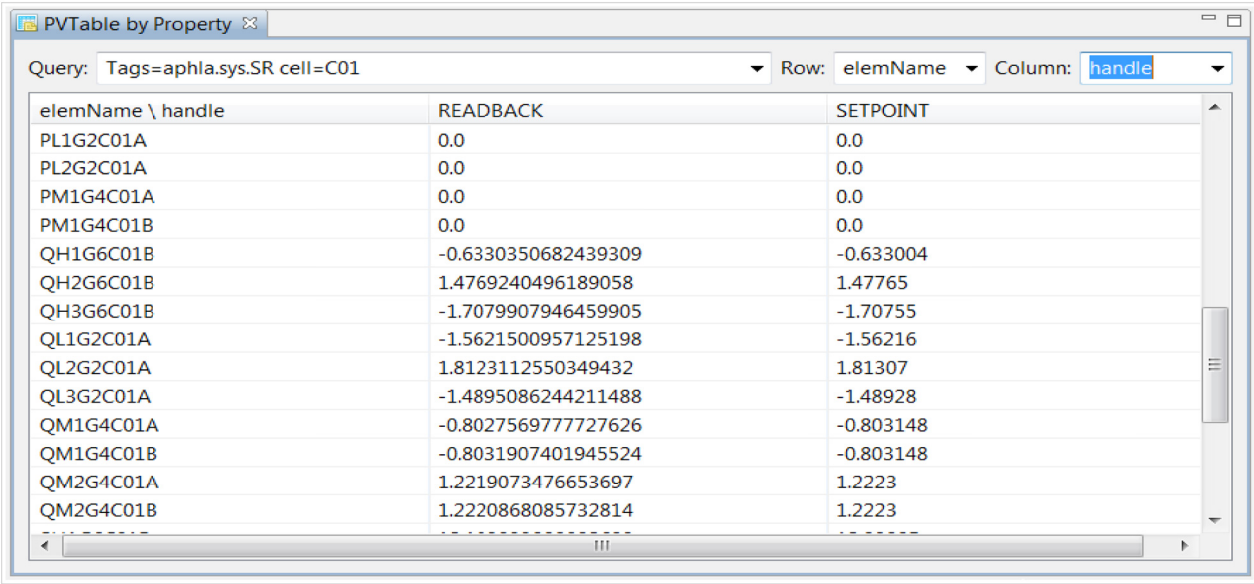


Figure 3: Shows an example use case of the MultiChannel Viewer, the application queries ChannelFinder for all the process variables associated with sextapoles(elemType=SEXT) in the storage ring (SR:*) and plots each channel ordered by the sEnd value.



elemName \ handle	READBACK	SETPOINT
PL1G2C01A	0.0	0.0
PL2G2C01A	0.0	0.0
PM1G4C01A	0.0	0.0
PM1G4C01B	0.0	0.0
QH1G6C01B	-0.6330350682439309	-0.633004
QH2G6C01B	1.4769240496189058	1.47765
QH3G6C01B	-1.7079907946459905	-1.70755
QL1G2C01A	-1.5621500957125198	-1.56216
QL2G2C01A	1.8123112550349432	1.81307
QL3G2C01A	-1.4895086244211488	-1.48928
QM1G4C01A	-0.8027569777727626	-0.803148
QM1G4C01B	-0.8031907401945524	-0.803148
QM2G4C01A	1.2219073476653697	1.2223
QM2G4C01B	1.2220868085732814	1.2223

Figure 4: Example of a full-width figure showing the distribution of problems commonly encountered during paper processing. This figure is labeled with a multi-line caption which has to be justified, rather than centered.

The result of this query, a group of channels, is then plotted with their order along the x-axis being determined by the "Order By" property selected in the drop box.

PVTable by Property

Figure 4 shows the PVTable by Property application. It allows user to organize the output of many channels in a table, based on the value of two properties, thus each cell of the table represents the current value of a single channel, the position of each channel is determined by the value of selected row and column properties. The list of channels and the properties used to create the rows and columns are obtained by querying the ChannelFinder.

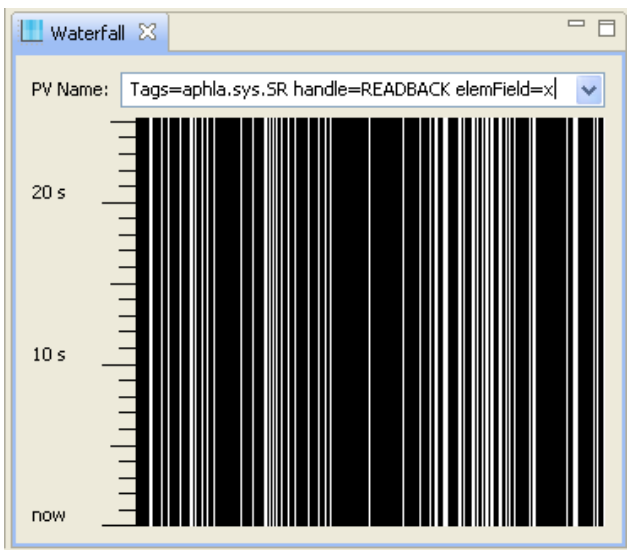


Figure 5: Waterfall plot generated from PVManager using the the result of the query for all channels with Tag=aphla.sys.SR and are readback values for element field x.

WaterFallPlot

The Waterfall plot shown in Figure 5. creates a time plot for all the values of a channel query or of a waveform.

CONCLUSION

- Use of the ChannelFinder service and client eliminates the need for configuring and remembering each individual channel.
- PVManager simplifies and improves performance of applications by handling various problems associated with use of large number of channel.

REFERENCES

[1] L. Dalesio, D. Dohan, R. Lange, A Proposal for introspection in EPICS, THP036, ICALEPS (2009).
[2] R. Lange, ChannelFinder: A directory service; http://channelfinder.sf.net.
[3] G.Carcassi, pvManager, EPICS Fall meeting (2010) ; http://pvmanager.sf.net
[4] Control System Studio; http://cs-studio.sf.net